

ソフトウェア工学の基本

1 導入

この講義で最も重要なのは、ソフトウェアは1回動けばよいのではなく、変化する要求の中で作り続け、直し続ける対象だということです。

情報工学ではアルゴリズムやデータ構造を学びますが、実際のソフトウェアでは、それをどう組み立て、どう壊れにくくし、どう保守するかが重要です。その視点を与えるのがソフトウェア工学です。

2 用語と定義

設計とは、ソフトウェアの構造や責務の分担を決めることです。

テストとは、ソフトウェアが期待どおりに動くかを確認する作業です。

保守とは、運用しながら修正や改善を続けることです。

3 方針

まず「なぜ設計が必要か」を整理し、そのあとテストと保守がなぜ重要かを見ます。

4 直感的な説明

小さなプログラムなら場当たりの書きでも動きますが、大きくなると変更の影響が広がります。だから、最初から「どこに何を置くか」「あとで直しやすいか」を意識する必要があります。

5 厳密な説明

5.1 1. 設計

設計では、機能をどう分割し、どの部品がどの責務を持つかを決めます。

5.2 2. テスト

テストは「動いたからよい」ではなく、「壊れていないことを確認する」ための仕組みです。

5.3 3. 保守

要求が変わると、ソフトウェアも変える必要があります。そのため、変更しやすい構造であることが重要です。

6 見分け方

- 実装だけでなく、変更やテストのしやすさを考える段階ならソフトウェア工学の話です。
- プログラムが大きくなるほど、設計と保守の重要性が増します。

7 最終形

ソフトウェア工学 = 設計 + テスト + 保守

8 一言でいうと

- ソフトウェア工学は、ソフトウェアを作るだけでなく育てていくための考えです。